**EXECUTION PROCESS**

*Created by*

*Sercan Şensülün and Emre KARAKIŞ*

*Submitted*

*Yrd.Doç.Dr Tuğkan Tuğlular*

**TABLE OF CONTENTS**

1.INTRODUCTION

2.EXECUTION PROCESS DESCRIPTION

1. **Project Creation**
2. **Import Existing Project To Eclipse:**
3. **Add External JAR files**
4. **Program Execution**
5. **Concrete Test Case Implementation**

3.CONCLUSION

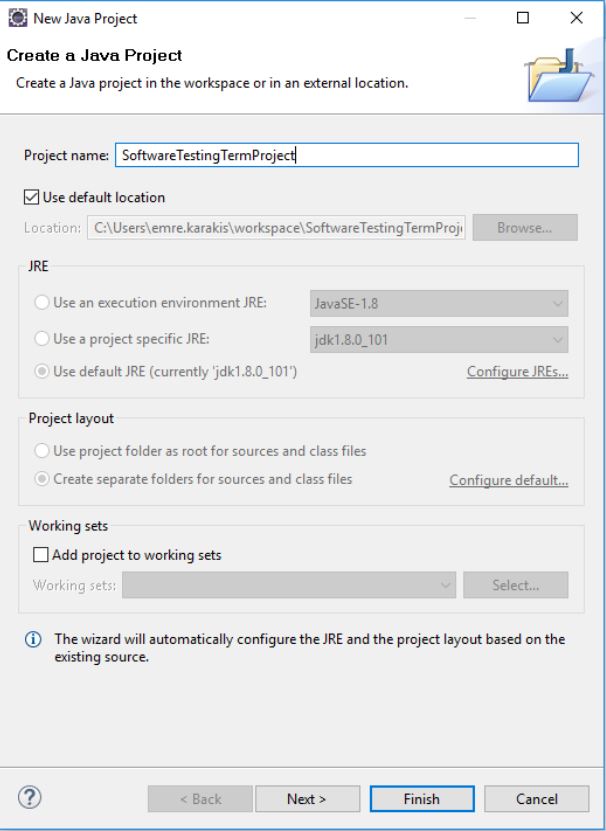
**1.INTRODUCTION**

In this assignment we tried to implement how abstract test cases depending on triangle inequality problem are automated. This implementation explains step by step how you can basicallyimport the project and execute the source code from Eclipse Mars 2 IDE and also adding necessary jar files to the existing project.

2.**EXECUTION PROCESS**

1. **Project Creation**

* In order to create a Java Project, initially open the Eclipse IDE and select **File -> New ->Java Project->** andthen enter a project name as shown in Figure 1.

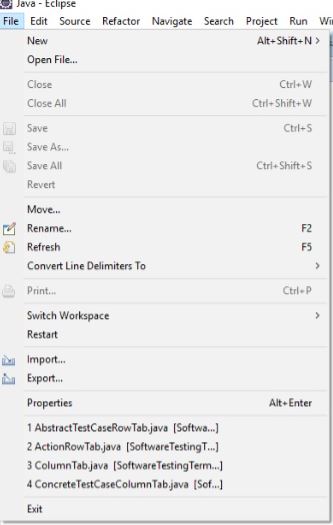


**Figure-1**

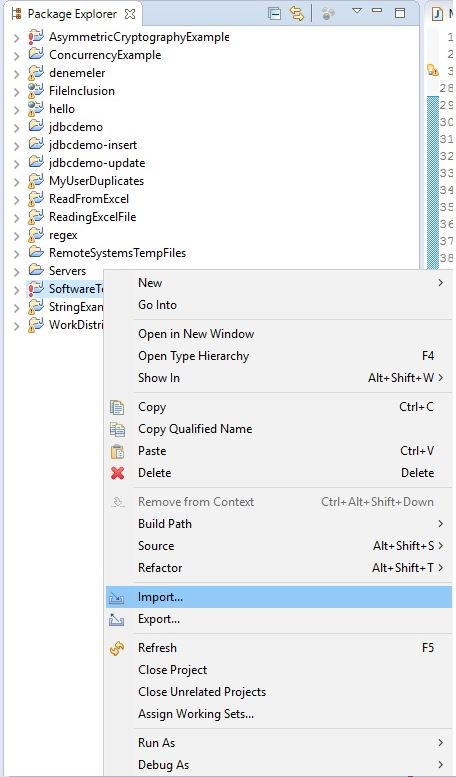
* After entering the project name, press finish button to display a Java Project in the Package Explorer segment.

**2.Import Existing Project To Eclipse:**

Select **File** and then **Import** as shown in the Figure-2.

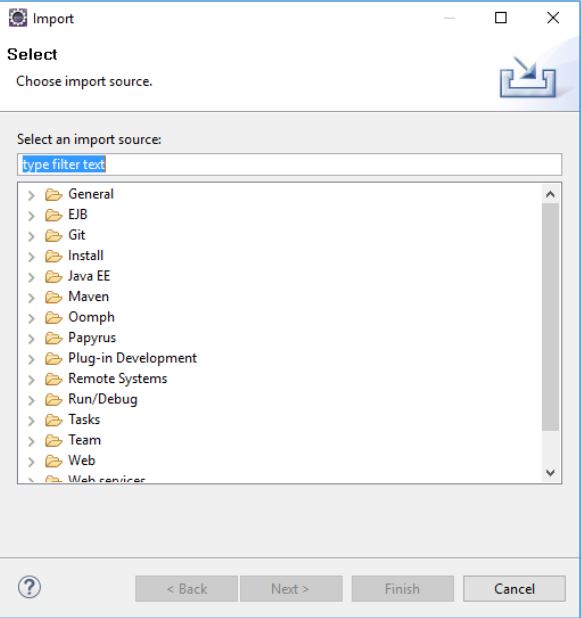


**Figure-2**



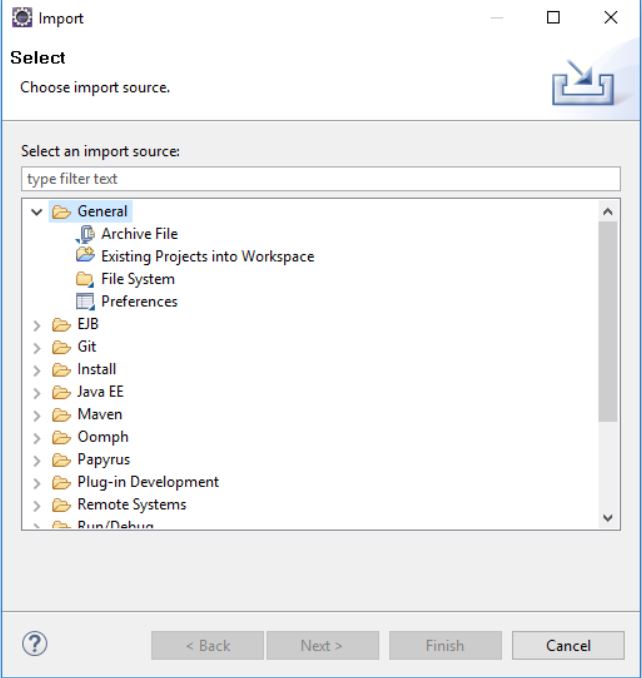
**Figure-3**

* After selecting Import, Figure-4 will be visible as shown below,



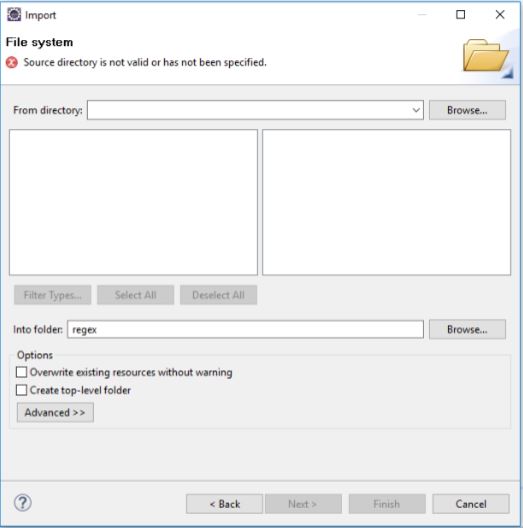
**Figure-4**

* After that, select **General**



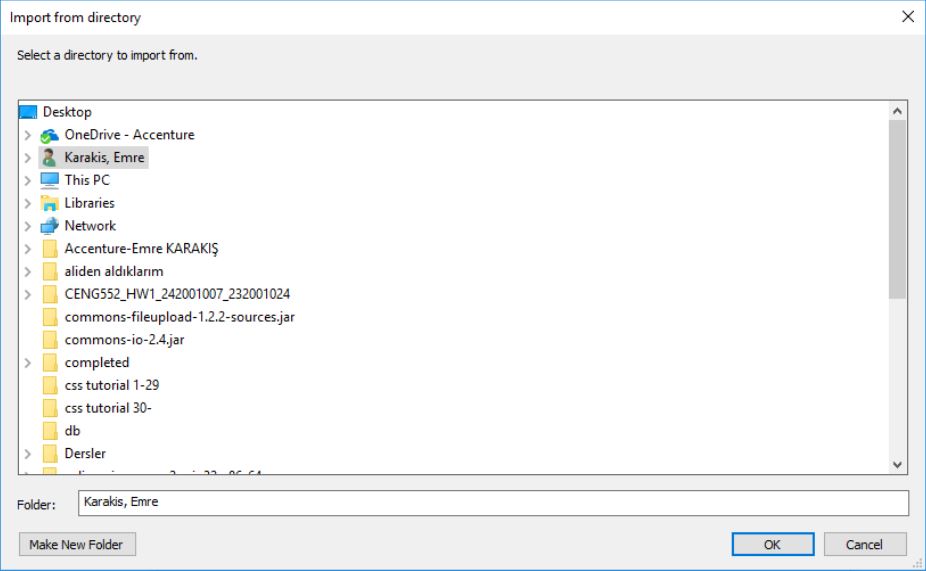
**Figure-5**

* After that **Select-> File System** and then find out the existing project from your file system by clicking on the **Browse** button.



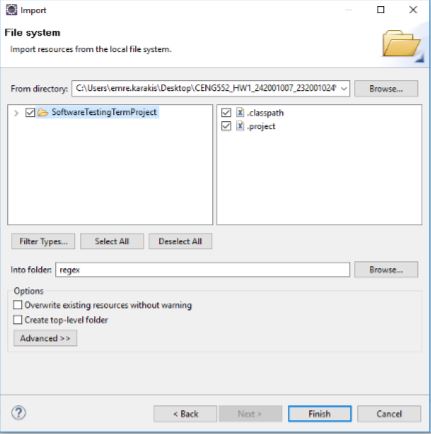
**Figure-6**

* The next step displays your file system like this



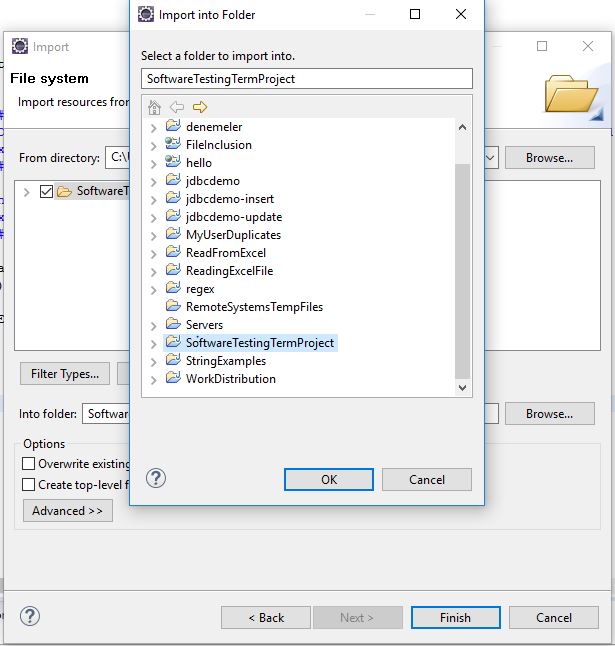
**Figure-7**

* Choose **CENG552\_HW1\_24001007\_232001024** file in your directory and select **SoftwareTestingTermProject** file and then **From directory segment** will be visible as shown below.



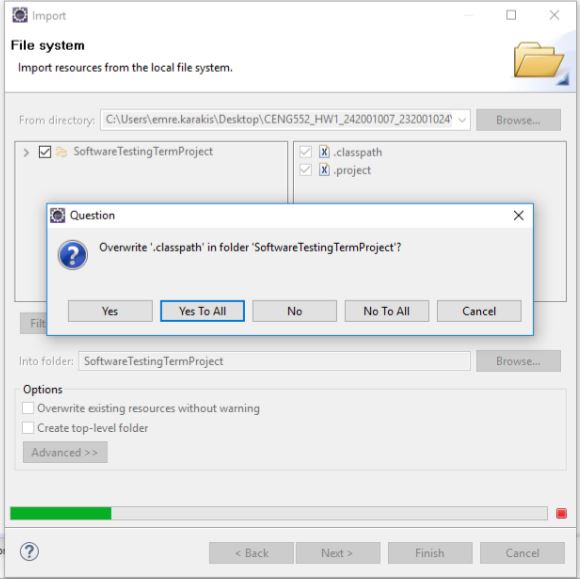
**Figure-8**

* In the following step , click **Browse** button in the **Into Folder** section as shown in Figure**.**Select the empty Java project which has previously created in the Eclipse Mars 2.0 IDE and named as “SoftwareTestingTermProject”.After choosing SoftwareTestingTermProjectfileand then click **Finish**.



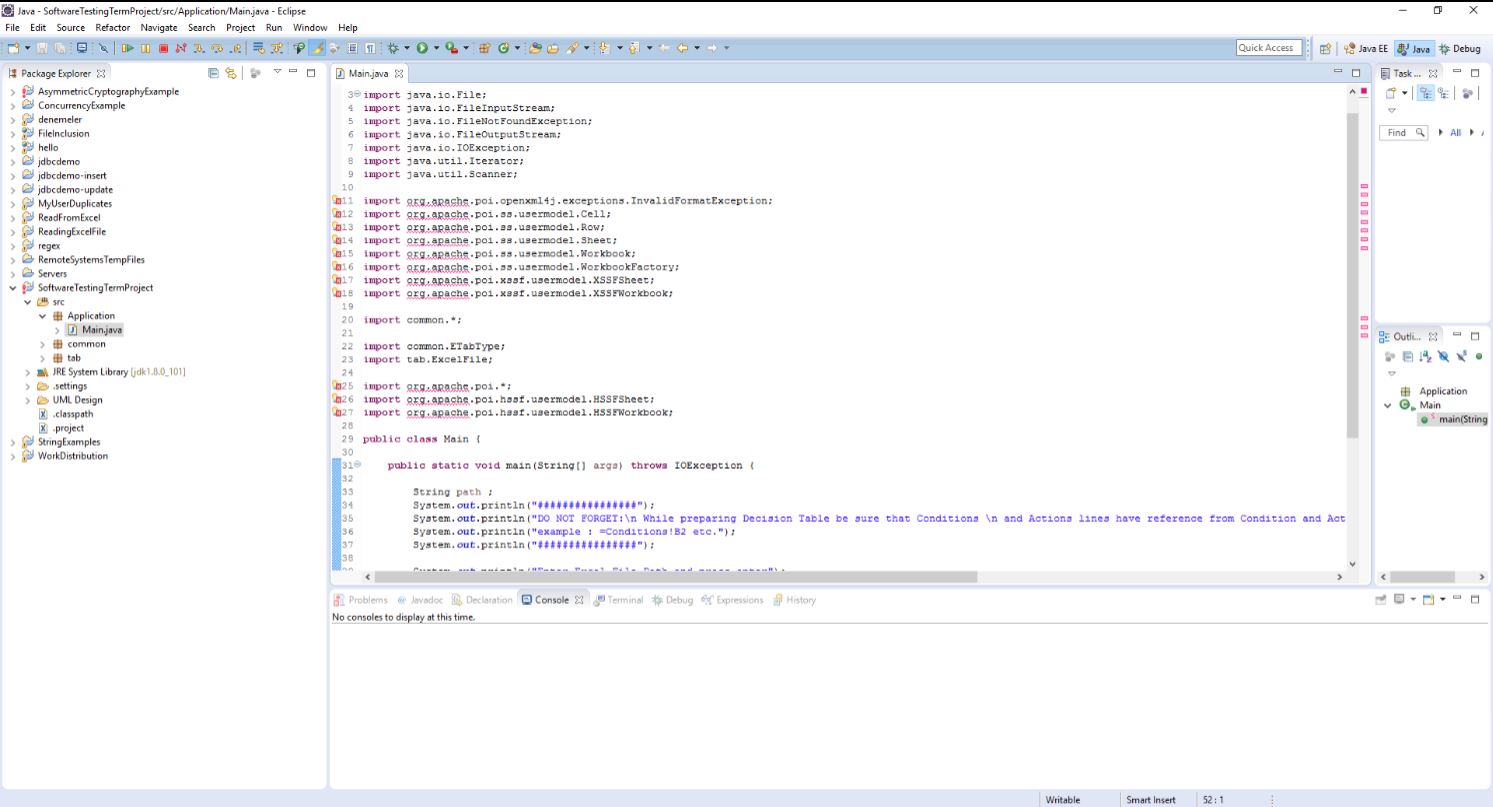
**Figure-9**

* After clicking **Finish** button, below figure will be visible.
* Click the **Yes to All** and then project will be successfully imported.



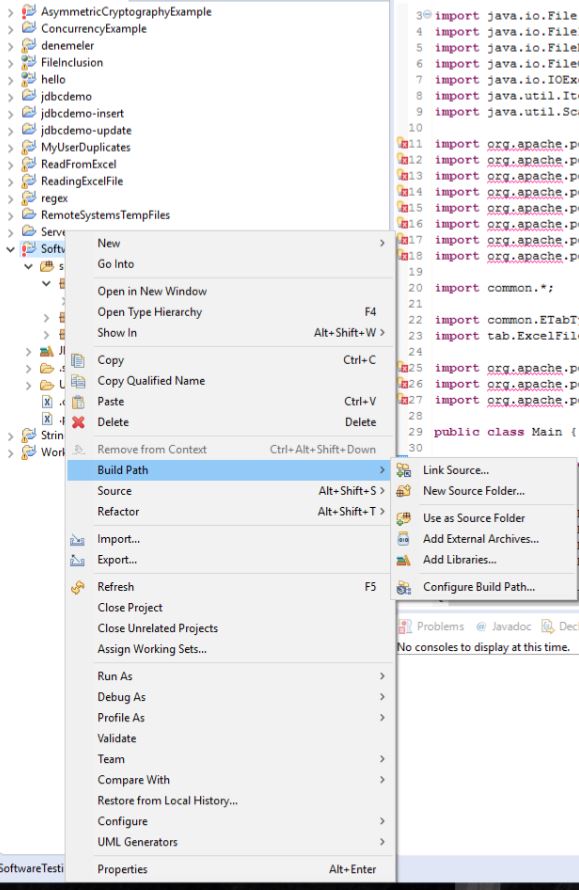
**Figure-10**

**3.Add External JAR files**

After importing the existing project in your directory, the project would contain some errors because of lack of jar files as shown below. In order to add jar files, first please go to **jar files** folder which is available under the **CENG552\_HW1\_24001007\_232001024** folder.

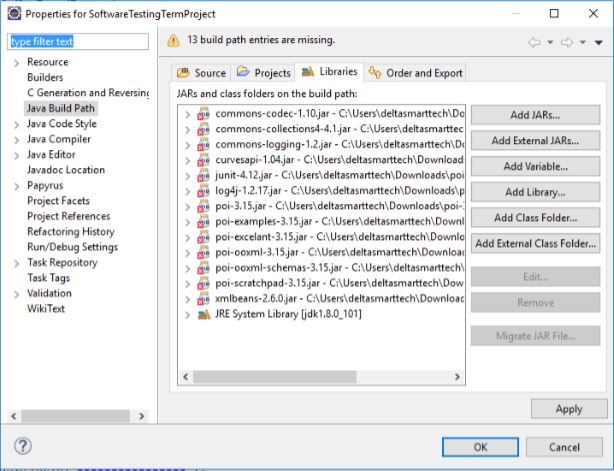
**Figure-11**

* In order to add jar files, please right click to the project named “SoftwareTestingTermProject” from **Package Explorer** and then select **Build Path -> Configure Build Path** as shown below.



**Figure-12**

* After selecting Configure Build Path, below figure will be visible.



**Figure-13**

* In that figure, click **Add External JARs** button, go to **jar files** folder which is given in project folder and select firstly and then click **Apply** and then **OK.**

**jar files\poi-3.15\poi-3.15.jar**

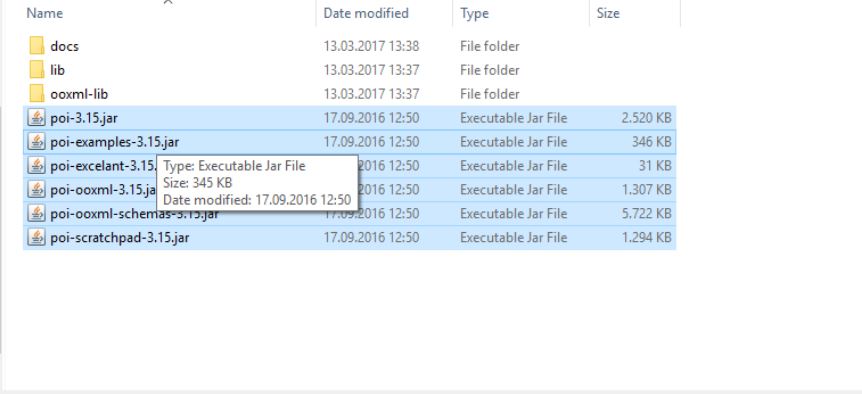
**jar files\poi-3.15\poi-examples-3.15.jar**

**jar files\poi-3.15\poi-excelant-3.15.jar**

**jar files\poi-3.15\poi-ooxml-3.15.jar**

**jar files\poi-3.15\poi-ooxml-schemas-3.15.jar**

**jar files\poi-3.15\poi-scratchpad-3.15.jar**



**Figure-14**

* After that, repeat the adding external jar process for selecting jar files under **jar files\poi-3.15\ooxml-lib\**

**curvesapi-1.04.jar**

**xmlbeans-2.6.0.jar**

* After that, repeatthe adding external jar process for selecting jar files under **jar files\poi-3.15\lib\**

**commons-codec-1.10.jar**

**commons-collections4-4.1.jar**

**commons-logging-1.2.jar**

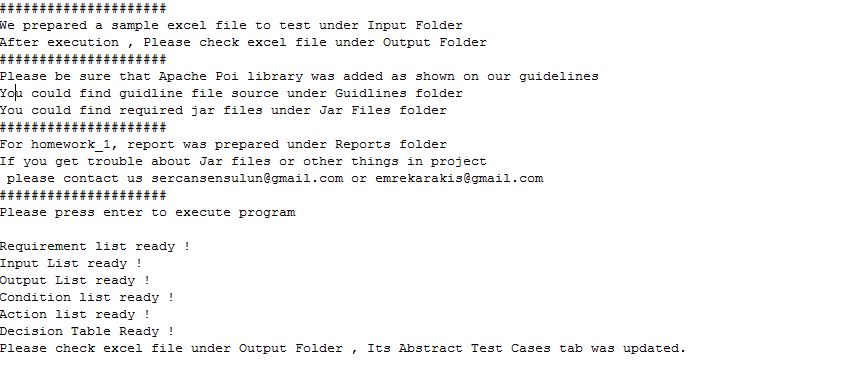
**junit-4.12.jar**

**log4j-1.2.17.jar**

By adding these jar files shown in above, all errors in Eclipse will disappear.

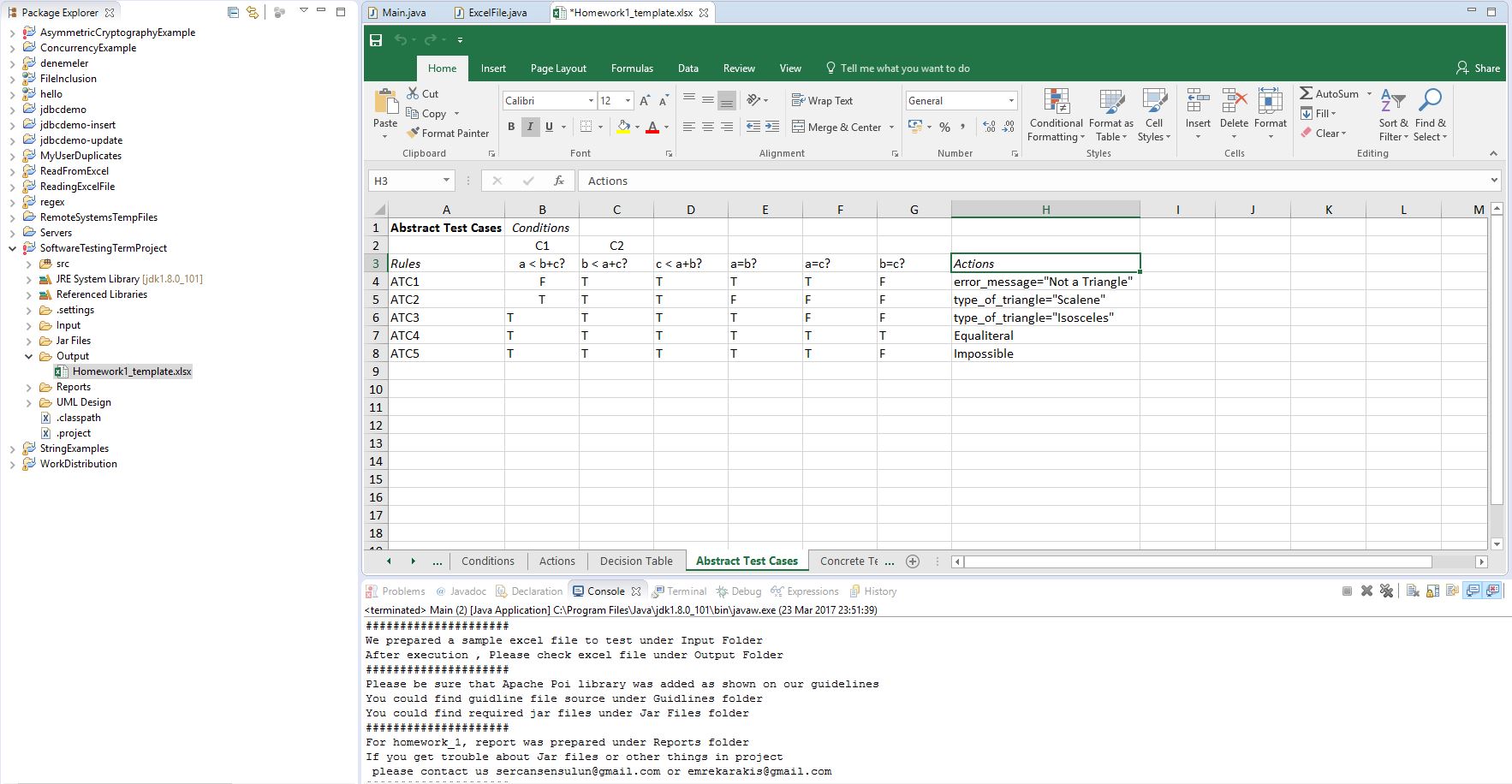
**4.Program Execution**

There are two specific folders in the project folder which is called Input and Output.We added a prepared excel file inside the Input folder.The program reads this input excel file and then generates the output excel file as soon as the user presses the “Enter” key.Thus, in order to execute program it is enough to press enter from keyboard. In that step, everything is available to execute source code from Eclipse IDE. Example execution would seem something like this.

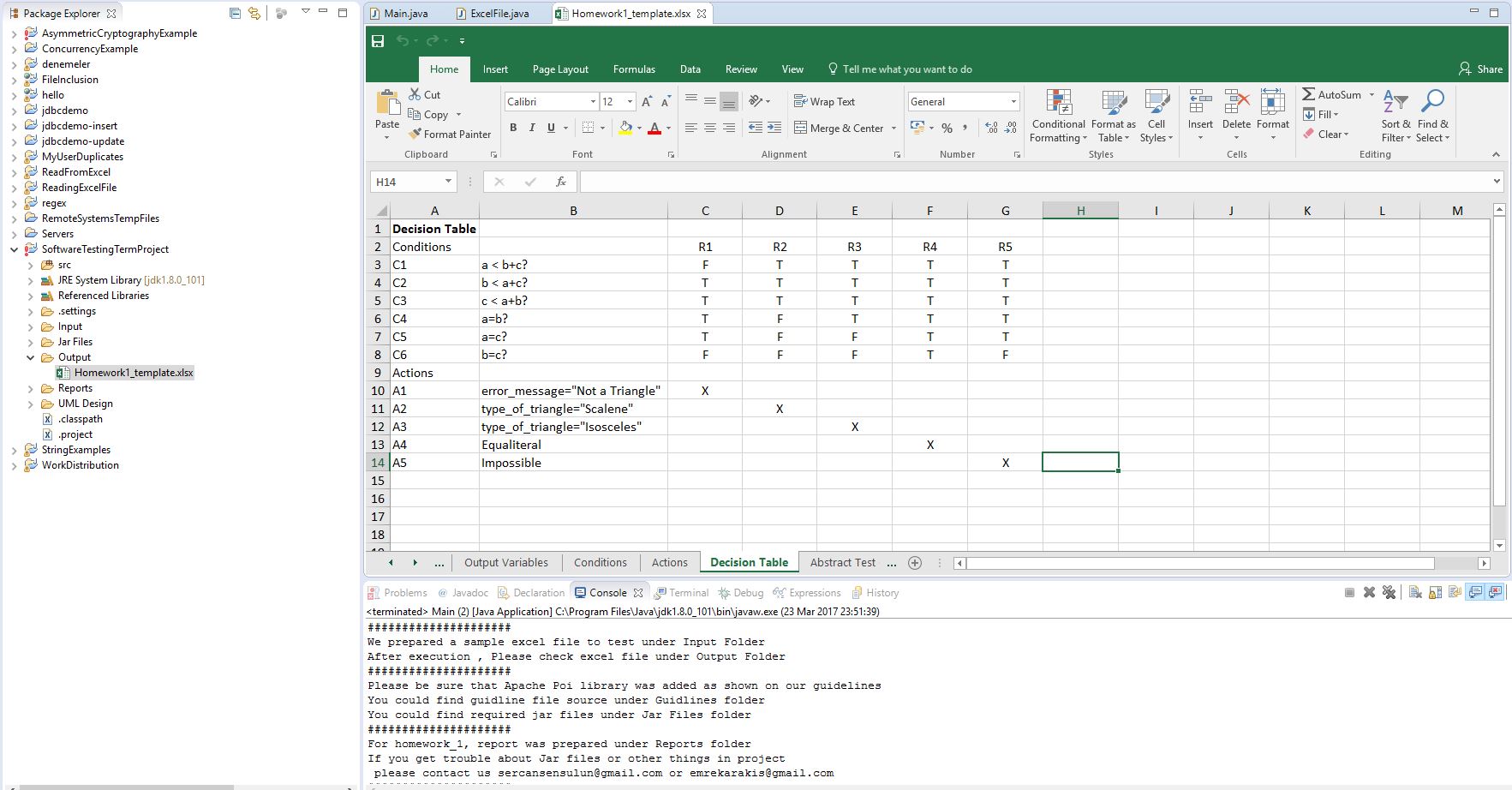


**Figure-15**

If you see this result, you can check the excel file inside the Output folder.The decision table tab would seem as shown in Figure 17 and Abstract test cases would be generated like Figure-18.



**Figure-16**



**Figure-17**

1. **Concrete Test Case Implementation**

For the concrete Test Cases, we did not implement sat4j.However, in the abstract test cases,we have previously generated 11 test cases, thus it is such a scenerio that we can optimize the test cases by filtering with the satisfiability condition.In some cases, abtstract cases are written then it does not reflect the actual point of view. Such inconsistencies would not assessed as a test case because these cases are impossible to consider, by that way we would have a little bit smaller list of test set.In homework 2, we add an additional implementation of Don’t Care(X) operations.When the user is asked whether a specific condition is satisfiable or not, then he/she answers the question for eleven test cases as Y/N or y/n.Then, concrete test cases are generated in the excel file within the output file in project.

**CONCLUSION**

As a consequence, in this documentation we tried to explain the basic steps of executions by giving input file to the system and then getting output file including abstract test cases which is automatically filled by the program to successfully implenment abstract test case model.